IN THE CLAIMS

1.   (**Currently Amended**) A method of accessing state from a configurable processor, the method comprising:

preparing a first state-accessing instruction stream based on a first user description of the configurable processor;

transmitting, using a debugger, [[a]] **the first** state-accessing instruction stream to an interpreting agent, the interpreting agent being capable of interpreting that stream; [[and]]

causing, using the **first** state-accessing instruction stream, the interpreting agent to return the state of [[the]] **a first configuration of the configurable** processor to the debugger

preparing a second different state-accessing instruction stream based on a second different user description of the configurable processor;

transmitting, using the debugger, the second state-accessing instruction stream to the interpreting agent, the interpreting agent also being capable of interpreting that second stream; and

causing, using the second state-accessing instruction stream, the interpreting agent to return the state of a second different configuration of the configurable processor to the debugger.

2.   (Original) A method as in claim 1 where the interpreting agent is a monitor program.

3.   (Original) A method as in claim 1 where the interpreting agent is an instruction insertion server.

4.   (Original) A method as in claim 1 where the interpreting agent is an architectural simulator.

5.   (**Currently Amended**) A method as in claim 1, ~~further comprising~~ wherein the steps of preparing the first and second state-accessing instruction streams include:

2

reading, using the debugger, information describing the **first and second** configurable processor's state architecture, **respectively**; and

generating, using the debugger, the **first and second** instruction [[stream]] **streams, respectively,** based on the information.

6.   (Original) A method as in claim 5 wherein the interpreting agent is a monitor program.

7.   (Original) A method as in claim 5 wherein the interpreting agent is an instruction insertion server.

8.   (Original) A method as in claim 5 wherein the interpreting agent is an architectural simulator.

9.   (Previously Presented) A computer-readable storage medium storing therein a software program comprising:

software for automatically generating a hardware description of a configurable processor from a user description of that processor; and

a debugger library for automatically generating information necessary to describe save and restore instructions for state of the configurable processor based on the user description.

10.   (Currently Amended) A computer-readable storage medium storing therein a debugger library for:

reading a description of save and restore state information of a configurable processor; and

generating saving and restoring state instruction streams **that are capable of being executed on the configurable processor** based on the description.

11.   (Currently Amended) A medium as in claim 10 wherein the debugger library further comprises functionality for:

3

identifying interdependencies in <u>a</u> state <u>that has been added to the configurable</u> <u>processor based on a user description</u>; and

generating a complete and correct save and restore sequence based on the interdependencies.

12.   (Currently Amended) An instruction-insertion server comprising:

<u>means for interpreting both first and second different state-accessing instruction</u> <u>streams, the first and second instruction streams being generated based on first and second</u> <u>different user descriptions of a configurable processor, respectively;</u>

means for retrieving system topology information of a chip containing multiple cores from a computer-readable file; and

means for determining where [[elements]] <u>first and second cores corresponding the</u> <u>first and second different user descriptions</u> are in a system described by the file; and

means responsive to the determining means for directing [[a]] <u>the first and second</u> state-accessing instruction stream to ~~an appropriate one of the multiple~~ <u>the first and second</u> cores, <u>respectively</u>.

13. (Currently Amended) A system for accessing state from a configurable processor, the system comprising:

a debugger which transmits a state-accessing instruction stream;

an interpreting agent which

receives the instruction stream,

interprets the instruction stream to access state of the configurable processor, and

returns the accessed state of the configurable processor to the debugger<u>,</u>

<u>wherein the interpreting agent is capable of interpreting both first and second ones</u> <u>of the state-accessing instruction streams which access state of first and second different</u> <u>configurations of the configurable processor, respectively, the first and second</u> <u>configurations being generated based on first and second different user descriptions of the</u> <u>configurable processor</u>.

14.   (Original) A system as in claim 13 where the interpreting agent is a monitor program.

15.   (Original) A system as in claim 13 where the interpreting agent is an instruction insertion server.

16.   (Original) A system as in claim 13 where the interpreting agent is an architectural simulator.

17.   (Previously Presented) A system as in claim 13, wherein the debugger is adapted to:
      read information describing the configurable processor's state architecture; and
      generate the instruction stream based on the information.

18.   (Original) A system as in claim 17 wherein the interpreting agent is a monitor program.

19.   (Original) A system as in claim 17 wherein the interpreting agent is an instruction insertion server.

20.   (Original) A system as in claim 17 wherein the interpreting agent is an architectural simulator.